SOCR 1.03: A Handwritten Data Form Producing and Reading System

Hanchuan Peng[1,2] and Qiang Gan[1,3]

[1]Department of Biomedical Eng.,
Southeast University, Nanjing, 210096, China
Email: phc@seu.edu.cn

[2]Department of Electronic & Information Eng.,
The Hong Kong Polytechnic University, Hong Kong.
Email: phc@eie.polyu.edu.hk

[3]Image, Speech and Intelligent Systems Research Group,
Department of Electronics and Computer Science,
University of Southampton, UK.
Email: qg@ecs.soton.ac.uk.

This paper is submitted for MMWS2000.

Category of this paper should be "Multimedia Applications".

Author in correspondence should be:

Dr. Hanchuan Peng
EIE,
The Hong Kong Polytechnic University,
Hong Kong.
Email: phc@eie.polyu.edu.hk

SOCR 1.03: A Handwritten Data Form Producing and Reading System

Hanchuan Peng[1,2] and Qiang Gan[1,3]

[1]Department of Biomedical Eng.,
Southeast University, Nanjing, 210096, China
Email: phc@seu.edu.cn

[2]Department of Electronic & Information Eng.,
The Hong Kong Polytechnic University, Hong Kong.
Email: phc@eie.polyu.edu.hk

[3]Image, Speech and Intelligent Systems Research Group,
Department of Electronics and Computer Science,
University of Southampton, UK.
Email: qg@ecs.soton.ac.uk.

Abstract **– Document analysis and recognition software, especially form reader, is greatly required in office automation. This paper serves as a primer to the SOCR (version 1.03), which is a software package for handwritten data form producing and reading. This package integrates techniques of database linking, form producing, form reading, and data verification. With SOCR, data forms can be easily created and the form data, including loosely constrained handwritten numerals and symbols, can be read into databases at high accuracy and high speed, and in a human-compatible manner. This software package has been applied to producing and reading handwritten student score forms in several universities and tax forms in several cities.**

Index Terms **– Document processing, form reading, pattern recognition**

I. INTRODUCTION

Automatic document processing is widely required in modern office work. There are two major categories of document processing. The first category is involved in occasional scanning of a small number of individual pieces of text, without major time constraints. The second aims at recognizing thousands of documents everyday, which usually needs highly qualified techniques for complex document structure analysis and Optical character recognition (OCR). One typical example of such applications is data form reading.

Dorronsoro, *et al* [1] concluded that two "contradictory" requirements have to be met for the second category of document processing:

(1) a not too high rejection rate, which should send back a large number of effects to manual processing,

(2) a very low error rate, which does not lead to costly and lengthy checking and remedying operations.

It is obvious that large-scale applications of document processing have imposed a very high standard on the OCR technology. In some representative cases, OCR is not enough to complete the reading task because a number of database operations are often involved. In addition, there should be some mechanisms developed to dynamically link the specific fields in forms to the required fields in database records. These mechanisms will further unavoidably require techniques of form matching/registration.

In recent years, we have developed a software package SOCR as an integrated solution to industrial applications of large-scale form reading. Besides many techniques of form reading, a module for form producing is also included in this software. This paper aims to give a brief introduction to SOCR (version 1.03). In section II, the main paradigm of SOCR is introduced. In section III, some implementation schemes of SOCR are explained. Section IV gives a short conclusion.

II. PARADIGM OF SOCR 1.03

The current version of SOCR, version 1.03, is a Win32 application package containing two major parts. One part is for form producing, while the other is for form data acquiring. Fig.1 illustrates the

main paradigm of SOCR 1.03. A completed process in using SOCR includes three steps:

(1) Users extract information from databases, which is going to be combined with form templates prepared in the module of form design, and print off the created forms.

(2) Users distribute forms and collect the filled data forms.

(3) The collected forms are read via SOCR again and stored into corresponding databases.

To attain good performance in data form reading, two classes of techniques are employed. One is for form reading, which consists of form registration, form decomposition, highly qualified handwritten OCR and Optical mark recognition (OMR), multi-form linking and rejection processing, *etc*. The other is data verification technique, which examines the reasonability of recorded data and corrects possible data errors.

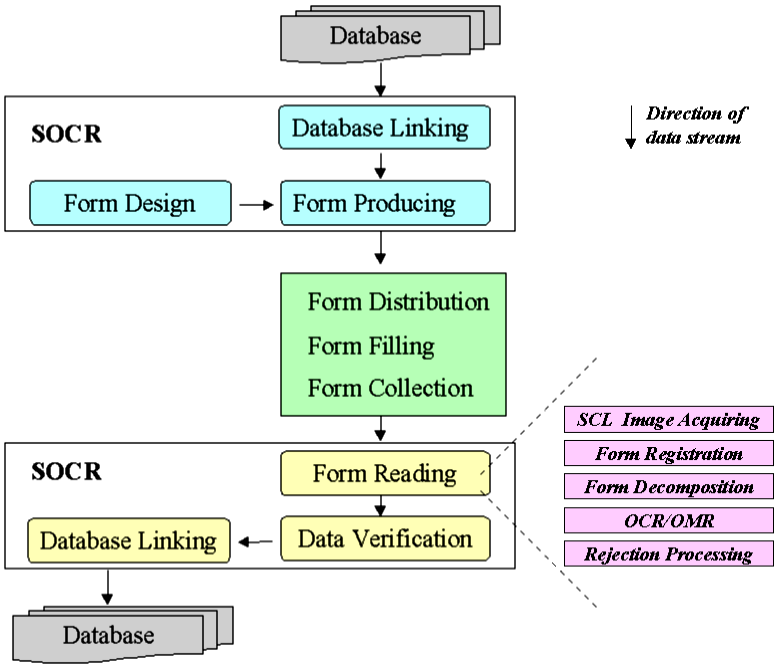


Fig.1 Paradigm of SOCR 1.03

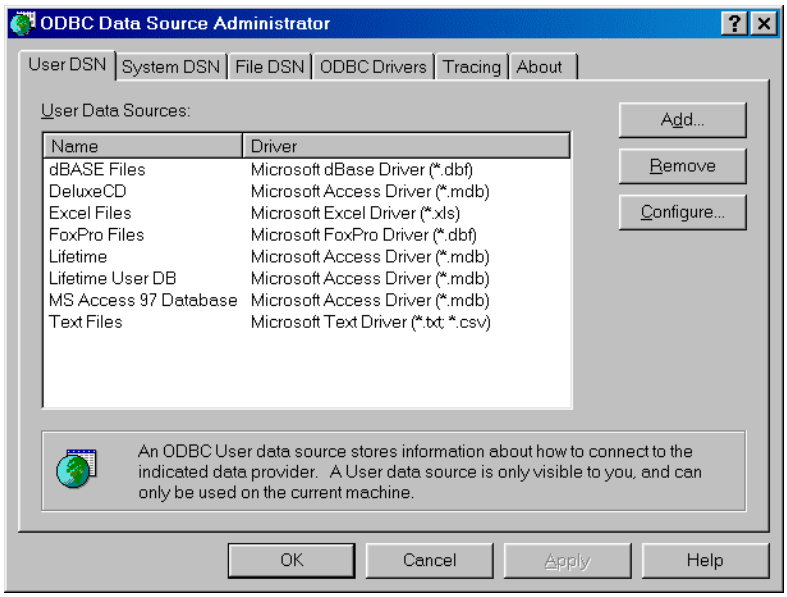


Fig.2 Database linking using ODBC, by which SOCR 1.03 can manipulate data in various databases.

III. IMPLEMENTATION OF SOCR 1.03

A. Database Linking

Database linking is an important feature of SOCR 1.03. With this function, SOCR can access existing databases and enable seamless usage of data forms in the databases and form data in paper documents. The database linking is realized with techniques of Open database connectivity (ODBC) and Data access objects (DAO), which make it easy to connect a variety of data sources (through the "ODBC" item in Windows control panel, as shown in Fig.2) and perform Structured query language (SQL) queries. In SOCR 1.03, the format of database records that should be used is defined in advance. However, it is possible for SOCR to dynamically define or choose record format.

B. Form Structure

Tree structure, as shown in Fig.3 (a), is used to describe and organize forms. A form is divided into several bodies, each having several rows. To obtain a stronger ability to describe various form formats, each row can be divided into several pillars, which are shown with the discontinuous connections in Fig.3 (a). Each pillar can have a multi-line structure, where each line constitutes a cell. However, for simple forms, a row can be directly separated into cells, which make up of the multi-column structure in the row. Examples of these two cases are shown in Fig.3 (c) and (b), respectively.

In Fig.3 (b), a simple form with three bodies is depicted. Each cell in these bodies is labelled with a *body-row-cell* string. In the first form unit in the third body, there is a separating slash. It will be ignored in the *body-row-cell* string labelling processing. Hence this form unit is labelled as B3R1C1. Obviously, the hierarchy of *body-row-cell* can effectively handle a large amount of form formats.

In Fig.3 (c), a complex form with three bodies is shown. A *body-row-pillar-cell* string is used to describe the logic position of a cell. In the first body of this form, cell B1R2P1C1 and cell B1R2P1C2 are obtained by virtually splitting a larger cell into two pieces. That is, the second row in this body is thought having four cells in total, although actually it just has three physical cells. In the second body of this form, cell B2R3P1C1 and cell B2R3P1C2 are thought as staying in the same row, together with cell B2R3P2C1LT and B2R3P2C1RB. Notice that four special describers, "LT" (left-top), "LB" (left-bottom), "RT" (right-top), "RB" (right-bottom) are used to label form units where there are separating slashes. Therefore, unlike the case in Fig.3 (b), the form unit B3R1P1C1 in the third body has two cells: B3R1P1C1LB and B3R1P1C1RT. Similarly, the more complicated cells such as B2R3P2C1LT and B2R3P2C1RB can be easily located by this labelling method.

If there is no ambiguity, the cell label can be shorten. For example, the cell B3R2P2C1 in Fig.3(c) can be shorten as B3R2C2, similar to the corresponding cell in Fig.3 (b).

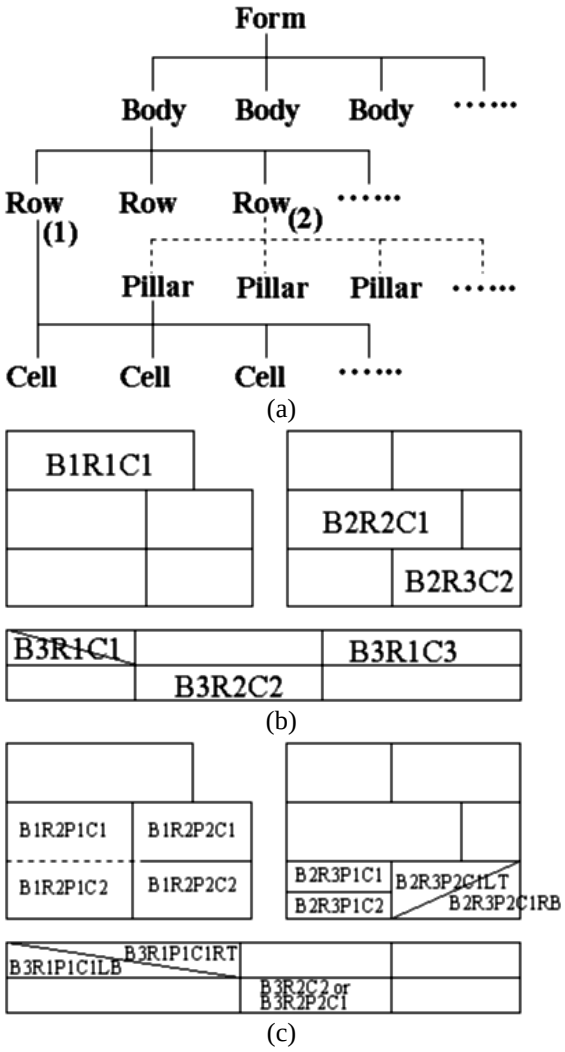


Fig.3 Form data structure. (a) The hierarchy tree, where the level of "Pillar" is optional, depending on applications. (b) An example of form without the "Pillar" level. (c) An example of form with the "Pillar" level, which can describe more complex structures.

It is noteworthy that for most applications, the *body-row-cell* string labelling may be enough to describe a variety of forms because information

extracted from ralational databases usually has well-estabilished row-column format.

C. Form Producing

In the form design module, form templates are prepared in advance or temporarily drawn. Different from the earlier versions of SOCR, in which the form producing is a function implemented by users, SOCR 1.03 provides functions to merge the data extracted from database and to link these data with specified fields of form templates. Users can preview (with functions of zoom in, zoom out, magnifying glass, *etc.*) details of ready-to-printing forms. Functions for refinement are provided to produce satisfactory hard copy forms. Fig.4 is the first pane of the user interface in the form producing wizard.

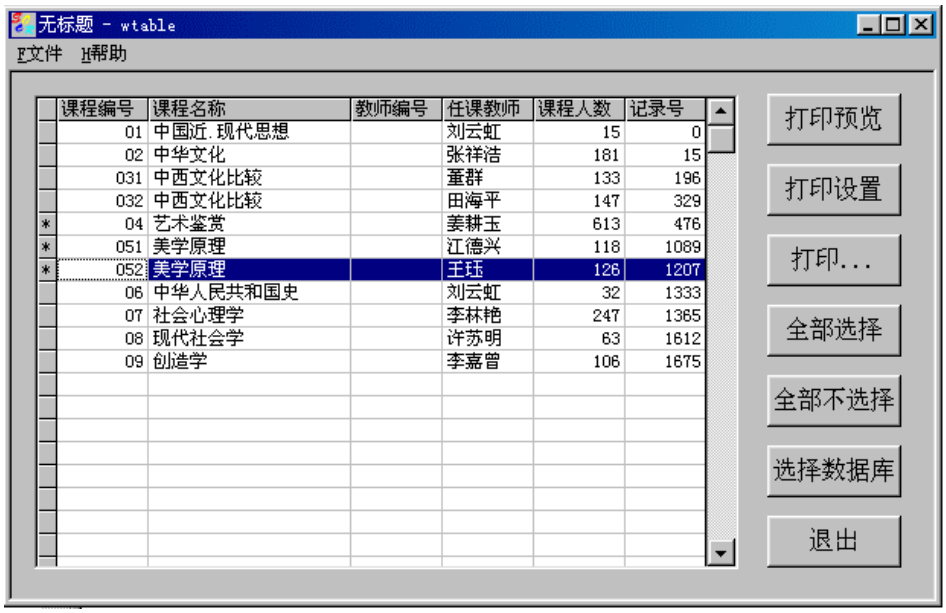


Fig.4 The first pane of user interface in form producing wizard. Users can use this wizard to choose data sources, to produce wanted data items, to control the quality of data forms, and so on.

D. Form Registration

After a form image is acquired by a page scanner, form registration is performed to decide which format the form is in. Form registration is based on the following techniques:

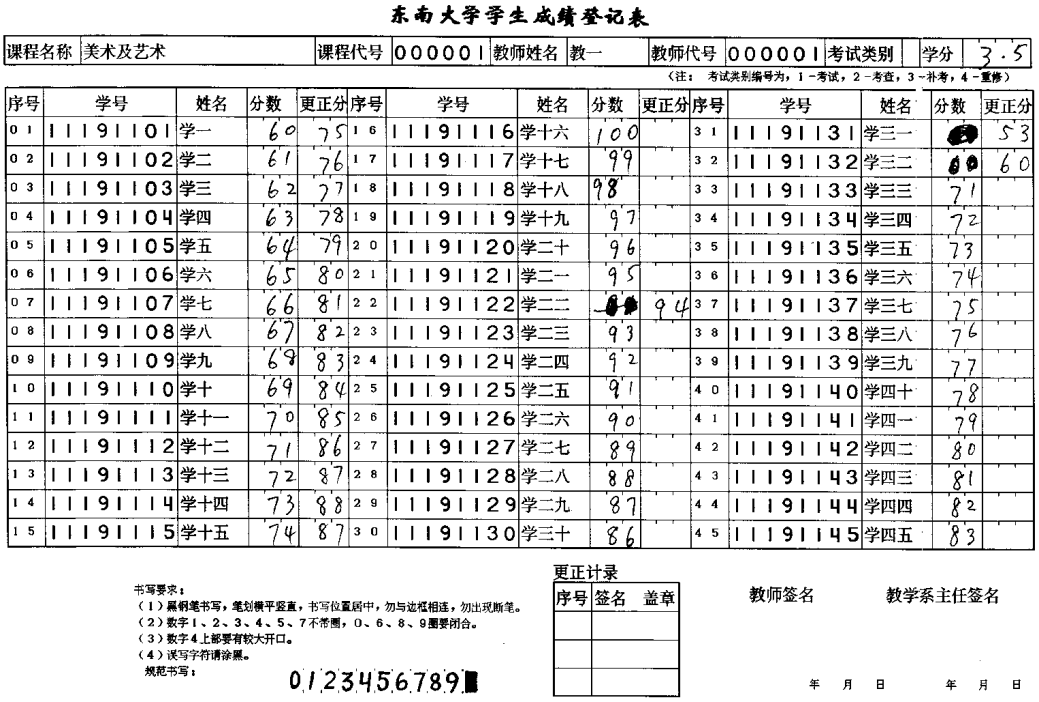
(1) A form learning technique is developed to scan a blank form and set meanings to all the cells in the form.
(2) A simple Form describing language (FDL) is created to record all format information of a form in a template file.
(3) Several methods are utilized to examine the format of a form. One method is feature matching based on key point detection. If the current form has the same key features with a form template, the form template is selected and meanings of all the form cells are decided according to the preset meanings in the template. Another method is bar code matching, in which different bar codes are printed on different types of forms in form producing. The third method is page matching, where the current form and the template form are accurately compared. The third method is time-consuming.

E. Form Decomposition

SOCR 1.03 is mainly applied to industrial data form reading, in which a large amount of forms are read piece by piece via a paper auto-feeder. In this special situation, there is little chance to get a form image with large skew angle. However, it is still very possible that the scanned form images have some distortion (less than 3°), which is harmful for further processing. SOCR 1.03 removes such distortion in the form decomposition module with a shift-compensation technique. This technique fractionates the globally large form distortion into locally small distortions. Because handwritten characters are to be recognized in SOCR, locally

small distortions can be considered together with the instability of handwritten characters. This approach has been proved rather realistic and satisfactory in several applications.

In the form decomposition module, images of cells are extracted and further separated as sub-cell character images, which are sent to the OCR/OMR module. For example, in the sample form of student score (Fig.5), the field of score has three characters, whose images are extracted from the corresponding cell image and sent to the recognition module.



东南大学学生成绩登记表

课程名称	美术及艺术	课程代号	000001	教师姓名	教一	教师代号	000001	考试类别		学分	3.5

（注： 考试类别编号为，1－考试，2－考查，3－补考，4－重修）

序号	学号	姓名	分数	更正分	序号	学号	姓名	分数	更正分	序号	学号	姓名	分数	更正分
01	11191101	学一	60	75	16	11191116	学十六	100		31	11191131	学三一	[illegible]	53
02	11191102	学二	61	76	17	11191117	学十七	99		32	11191132	学三二	60	60
03	11191103	学三	62	77	18	11191118	学十八	98		33	11191133	学三三	71	
04	11191104	学四	63	78	19	11191119	学十九	97		34	11191134	学三四	72	
05	11191105	学五	64	79	20	11191120	学二十	96		35	11191135	学三五	73	
06	11191106	学六	65	80	21	11191121	学二一	95		36	11191136	学三六	74	
07	11191107	学七	66	81	22	11191122	学二二	[illegible]	94	37	11191137	学三七	75	
08	11191108	学八	67	82	23	11191123	学二三	93		38	11191138	学三八	76	
09	11191109	学九	68	83	24	11191124	学二四	92		39	11191139	学三九	77	
10	11191110	学十	69	84	25	11191125	学二五	91		40	11191140	学四十	78	
11	11191111	学十一	70	85	26	11191126	学二六	90		41	11191141	学四一	79	
12	11191112	学十二	71	86	27	11191127	学二七	89		42	11191142	学四二	80	
13	11191113	学十三	72	87	28	11191128	学二八	88		43	11191143	学四三	81	
14	11191114	学十四	73	88	29	11191129	学二九	87		44	11191144	学四四	82	
15	11191115	学十五	74	87	30	11191130	学三十	86		45	11191145	学四五	83	

书写要求：
（1）黑钢笔书写，笔划横平竖直，书写位置居中，勿与边框相连，勿出现斯笔。
（2）数字1、2、3、4、5、7不带圈，0、6、8、9圈要闭合。
（3）数字4上部要有较大开口。
（4）误写字符请涂黑。
规范书写：0123456789

更正计录

序号	签名	盖章

教师签名　　教学系主任签名

年 月 日　　年 月 日

Fig.5 A sample form for student score collection. This format of form is in use by Southeast University and other two universities in China. The form format can be changed by the form producing module.

F. OCR/OMR

In SOCR 1.03, the OCR/OMR module can attain recognition accuracy around 99.5% on loosely constrained (handprinted) handwritten numerals and symbols. "Loosely constrained" just means that all the characters should not be indeterminate for classification. The constraints imposed on handwritten numerals include:

(1) '0', '6', '8', '9' have closed circles,
(2) '1', '2', '3', '4', '5', '7' do not have circles,
(3) '4' does not close the upper part.

Although recognition machines for much more unconstrained scripts have been included in SOCR 1.03, it is still strongly encouraged that form filler can follow these "loose constraints". From the report of practical use of SOCR in the past several years, it was found most users would not feel inconvenient for these writing constraints and could well keep the writing speed. In Fig.5, many typical handwritten numerals are shown.

The recognition engine of SOCR 1.03 is mainly based on a hybrid model, which combines traditional methods of numeral recognition and neural network recognizers. To decrease the error rate as minor as possible, SOCR will send all doubtful characters to rejection processing module.

G. Rejection Processing

In SOCR 1.03, a manual tool for rejection correction is integrated. The basic framework of this tool is shown in Fig.6. The first column and third column are the scanned images of form cells, which have bad qualities due to poor handwriting or scanning noise. The second column and the fourth column in Fig.6 are results directly from the OCR engine, where a cross "x" represents a rejection. Through this rejection correction tool, users can directly correct rejected characters (crosses) according to those cell images displayed on the left. Another major technique for the rejection processing is called multi-form linking: there is only

one rejection processing page for all the forms, which can distribute the corrected results back to the original forms where the specific rejection item emerges.

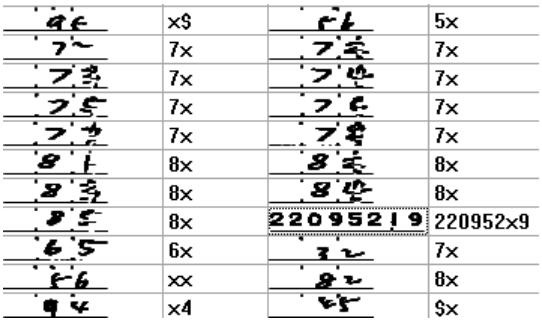


Fig.6 Framework of rejection correction

H. Data Verification

Before the recognized data are written to a database, their correctness and effectiveness are examined. SOCR employs different data verification strategies for various applications. In the application of student score reading, effective bits of student identification number and student score are examined, as well as their values. For example, usually the student score can at most have 3 digits and is less than 100. A pair examination for cell image and recognition result is also performed, as shown in Fig.7, where the handwritten character images and the recognized results can be compared. For the application of tax form reading, SOCR mainly investigates the reasonability and effectiveness of fields of money amount. Actually, all data verification methods in SOCR1.03 were discussed and decided together with customers and users.

IV. Discussion and Conclusion

SOCR 1.03 is developed for industrial applications of form producing and reading. With the function of database linking, SOCR can easily handle large-scale form processing tasks. Functions of form describing, producing, registration, and decomposition offer powerful ability to support simultaneous input of forms with tens of different formats. OCR/OMR module is implemented for the goal of high-accuracy recognition and doubtful character rejection. Automation of these processes results in high speed of form reading. Useful post-processing functions, including rejection correction, data verification, *etc*, are also included in SOCR 1.03. This software package has been appraised as a well-developed data form producing and reading system in China [2].

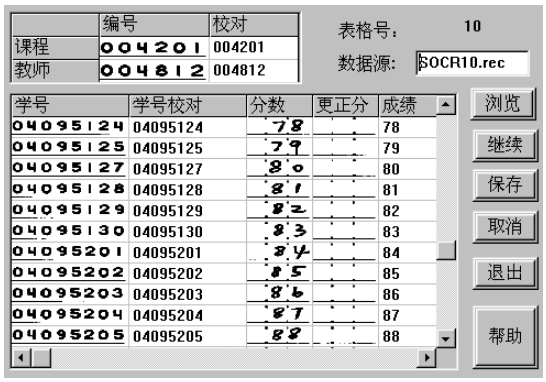


Fig.7 Example of manual data verification in student score reading

Acknowledgement

This work was supported by the Young Scientist Foundation from Jiangsu Province, China and by several industrial contracts. Part of this work (1994-1996) won a certificate of scientific/technical accomplishment awarded by the Committee of Science/Technology, Jiangsu Province, China [2]. Part of this work (1997) won the first grade prize in the 1997 Computer Accomplishment Competition of Chinese Universities. The authors would like to thank Jie Zhou* for the design and implementation of partial OCR algorithms used in this system. The authors also thank Jiangming Xie, Hao Chen, Lifeng Sha, Xuedong Zhang and Minghua Xu for their contributions to this work.

References

[1] J. Dorronsoro, G. Fractman, and C. Santa Cruz, "Large scale neural form recognition," in F.F. Soulie and P. Gallinari (Eds.) Industrial Applications of Neural Networks, World Scientific Publishing, pp.355-362, 1998.

[2] Q. Gan, H.-C. Peng, *et al*, "Research on auto-reading systems of handwritten data forms," Certificate of Scientific/Technical Accomplish-ment (awarded by the Committee of Science & Technology, Jiangsu Province, China), No. SKJ1996-331, Dec., 1996.

* Jie Zhou is currently working for her Doctoral Degree in Center for Pattern Recognition and Machine Intelligence at Concordia University, Canada. Her email is jzhou@cenparmi.concordia.ca.